A Lightweight Link Layer for BoW Interfaces

EMPOWERING OPEN.

OCP GLOBAL SUMMIT

OCTOBER 18-20, 2022
SAN JOSE, CA

A Lightweight Link Layer for BoW Interfaces

OPEN DOMAIN SPECIFIC ARCHITECTURE

David Kruckemyer, Principal Engineer, Ventana Micro Systems
Elad Alon, CEO, Blue Cheetah Analog Design

OPEN COMMUNITY®

Agenda

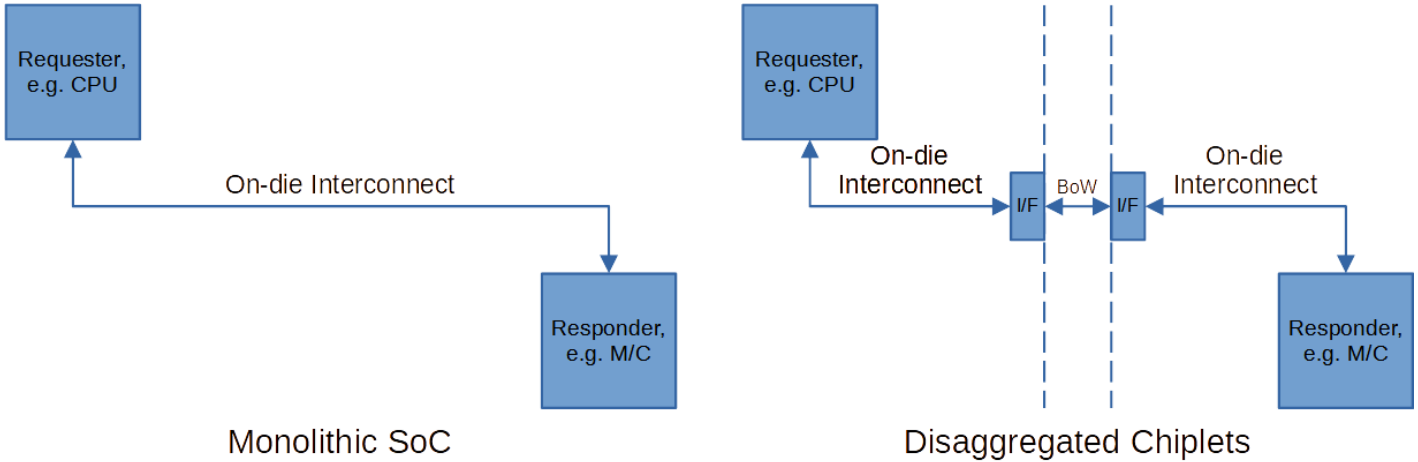
- Goals
- D2D Interface Architecture
- Transaction Layer
- Interface Profiles
- Link Layer
- Error Correction
- Training
- Virtual Wires

Goals

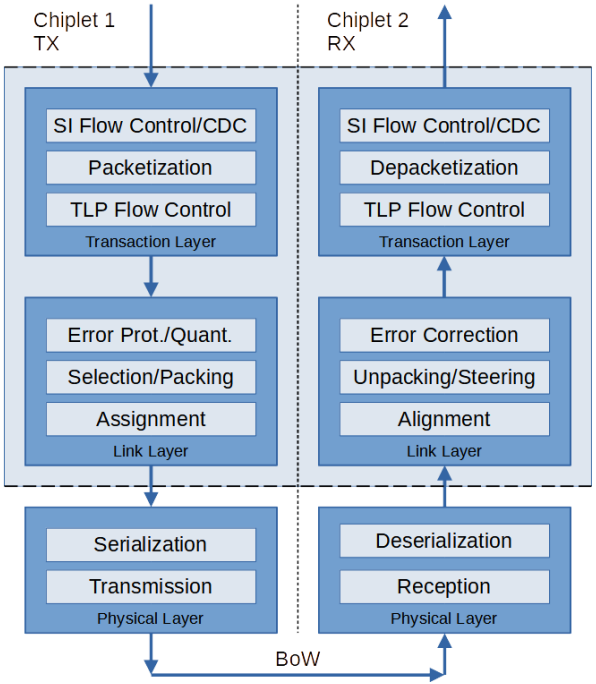
- **Simplicity** – build on the BoW physical layer; focus on die disaggregation; eliminate complexities of CDR, CRC/retry, etc.
- **Low Latency** – enable aggressive implementation techniques; use FEC to eliminate serialization overhead of CRC
- **Scalability** – support for different lane data rates and numbers of slices
- **Portability** – enable interfaces to be built with different implementation methodologies and in different process nodes
- **Extensibility** – create a modular framework that can easily add features over time; enable customization and interoperability via profiles

Die Disaggregation Model

- D2D interface implements standard system interface to ease IP integration
- Transparent to requesters and responders



D2D Interface Architecture



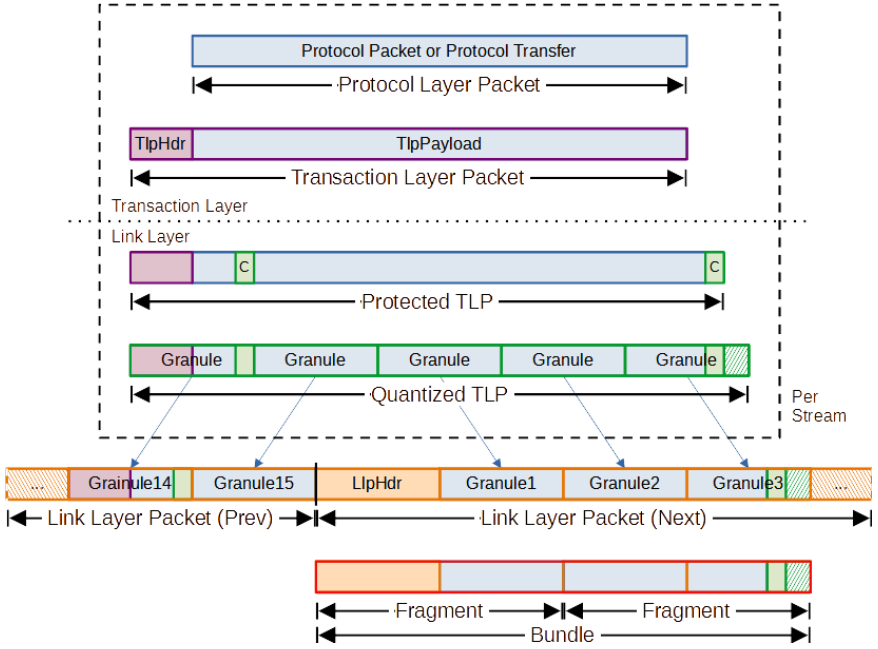
- Transaction Layer
  - Implements interface to system
  - Packetizes/depacketizes interface signals to TLPs
  - Manages credit-based flow control per TLP stream
- Link Layer
  - Protects TLPs with SECDED ECC
  - Packs/unpacks TLPs into/from LLPs
  - Trains/aligns fragments
- Physical Layer
  - Implements BoW slice logic interface
  - Supports 1, 2, or 4 BoW slices

Transaction Layer Functions

- TX Transaction Layer
  - Act as interface protocol responder including flow control and CDCs
  - Transform protocol transfers into transaction layer packets (TLPs), classifying the TLPs into streams and classes
  - Manage flow control for each stream with the far-side RX interface
- RX Transaction Layer
  - Manage flow control for each stream with the far-side TX interface
  - Transform TLPs into protocol transfers
  - Act as interface protocol requester including flow control and CDCs

Transaction Layer Packets

- Protocol packets (or transfers) are encapsulated into TLPs
  - 12b header with Type and Aux
- System interface supports various interface protocols and channels
  - Interface protocol → TLP class
  - Interface protocol channel → TLP stream



Interface Profiles

- A D2D controller that implements an interface profile is interoperable with another D2D controller that implements the same interface profile
- Interface profiles define various properties of the transaction layer
  - The inclusion or exclusion of optional fields on the interface protocol channels
  - The width of each field on the interface protocol channels
  - The combining and mapping of fields into the payloads of TLP types
  - The assignment of TLP types into TLP streams
  - The bundling of TLP streams into TLP classes
- Available interface profiles
  - Ventana VT-1, CHI-E RN-256, various AXI profiles, various AXI-Lite profiles, etc.
  - Designers allowed to develop bespoke profiles to meet application needs

Flow Control

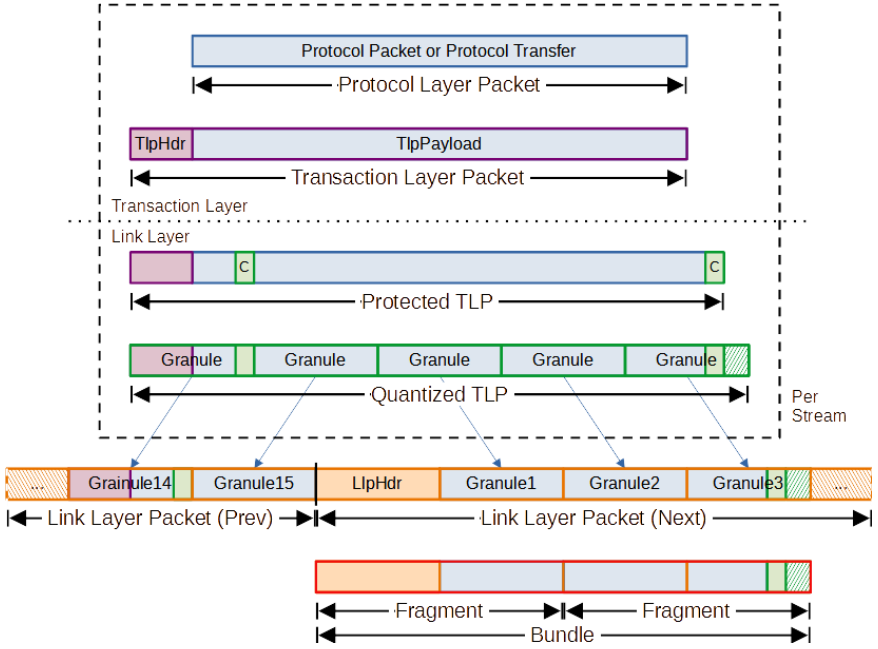
- The TX and RX interfaces manage each TLP stream with credits
  - Each credit represents an available TLP buffer in the RX interface
  - The TX interface requires a credit to transmit a TLP
  - A TLP for a given stream is only selected if a credit is available, ensuring non-blocking transfers for different streams
- Two methods are supported
  - Aux field in TLP header to maintain flow control bandwidth when link utilization is high
  - Credit TLPs to reduce latency when link utilization is low
  - TLP class determines the format of the above based on transaction structure

Link Layer Functions

- TX Link Layer
  - Add error-correcting code (SECDED) and quantize TLPs into granules
  - Select and pack eligible TLPs from different TLP streams into link layer packets
  - Assign LLP fragments to PHY slice logic interfaces
  - Drive the link layer training pattern
- RX Link Layer
  - Align LLP fragments received from different PHY slice logic interfaces
  - Unpack TLPs from LLPs
  - Correct single-bit errors and detect double-bit errors in TLPs
  - Sample the link layer training pattern

Link Layer Packets

- ECC is applied to TLPs
  - Header protected with small codeword
  - Payload protected with large codeword(s)
- TLPs are quantized into 32b granules
  - End is padded with zeros
- LLP is 512b
  - 32b header (1 granule) framing
  - 480b payload (15 granules)
- TLP granules are packed into LLPs and mapped into fragments and bundles

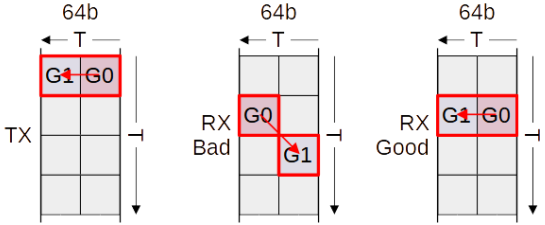


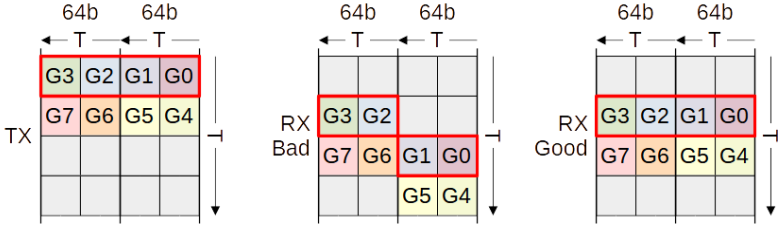
Error Detection and Correction

- Two SECDED ECC codewords
  - 32b “Header” codeword (26 data bits + 6 check bits)
  - 128b “Payload” codeword (120 data bits + 8 check bits)
- Small codewords fit into a single granule
  - Received atomically (vs. large codewords)
  - Framing information decoded and corrected inline quickly
- Large codewords provide better data transfer efficiency for payload
- Extremely reliable: p(uncorrected error) < 0.001 in 1B hours
  - BER = 10^{-15} @ 2 Tbps (8 slices @ 16 Gbps)

Training

- The TX interface transmits a known pattern, and the RX interface adjusts intra-slice phase and inter-slice skew based on the received pattern
  - The physical layer performs lane training
- Intra-slice phase errors are due to serialization and deserialization of fragments





- Inter-slice skew errors are introduced by variable delays between slices
  - Deskew buffer between the physical layer and link layer

Virtual Wires

- Virtual wires enable physical wires to be transported over the D2D link
  - Level-sensitive semantics, e.g. interrupts, status signals, etc.
  - More sophisticated signaling can be built with these primitives
- Low bandwidth overhead
  - A virtual wire TLP is transmitted when an input wire transitions on the TX interface
  - An output wire on the RX interface transitions when a virtual wire TLP is received
- Up to 1024 wires supported
- An interface profile specifies the number and definition of virtual wires

Conclusion

- A simple, low latency, scalable, portable, and extensible transaction and link layer framework architected for BoW interfaces
- Optimized for die disaggregation applications
- Interoperability and customization achieved through interface profiles

How to Participate/Call to Action

- ODSA Link Layer Working Group Meets on Thursdays at 9AM Pacific Time
  - Meeting link – https://global.gotomeeting.com/join/215559957
- Current work
  - Publishing additional interface profiles for various interface protocols
  - Developing sideband control interface
- Where to find additional information
  - GitHub repo – https://github.com/opencomputeproject/ODSA-Link-Layer

Thank you!

EMPOWERING OPEN.

OCP GLOBAL SUMMIT

OCTOBER 18-20, 2022
SAN JOSE, CA